



درس طراحی زبان‌های برنامه‌سازی

دکتر محمد ایزدی

کوییز اول

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۱۴۰۰-۱۳۹۹

مهلت زمان امتحان:

۵۰ دقیقه



به موارد زیر توجه کنید:

- * مدت زمان امتحان ۵۰ دقیقه است. پس از ۵۰ دقیقه امکان ارسال جواب در کورا وجود نخواهد داشت.
- * جواب تمام سوال‌ها را به صورت دست‌نوشته و یا تایپی بنویسید. سپس تمام جواب‌ها را در یک فایل pdf با نام دلخواه گذاشته و در قسمت مربوطه در کورا آپلود کنید. دقت کنید که در کورا تنها امکان ارسال pdf وجود دارد.
- * تنها ارسال نهایی شما در کورا تصحیح خواهد شد. دقت کنید که در صورت ارسال چند جواب، کورا به صورت خودکار آخرین ارسال را به عنوان ارسال نهایی در نظر می‌گیرد.
- * لطفا جواب سوال‌ها را از یکدیگر کپی نکنید. در صورت وقوع چنین مواردی مطابق با سیاست درس رفتار می‌شود.



سوال ۱

روش JIT (*Just in time compilation*) را به اختصار تعریف کنید و آن را با سایر روش‌های طراحی کامپایلر یا مفسر از نظر سرعت و قابلیت انتقال (*portability*) مقایسه کنید.

سوال ۲

در زبان c و c++ در هر یک از چهار عبارت زیر، تعریف f چیست؟ تفاوت آن‌ها با هم چیست؟

- `int f(int *x)`
- `int *f(int x)`
- `int (*f)(int x)`
- `int *f(int *x)`

سوال ۳

در هر مورد زیر مشخص کنید قوانین، مشخص‌کننده‌ی چه مجموعه‌ای هستند؟ درخت استخراج (derivation) را برای (6, 7, 13) برای مورد دوم بکشید.

- $(1, 2) \in S, \frac{(n,k) \in S}{(n+2,k+8) \in S}$
- $(0, 1, 2) \in S, \frac{(n,m,k) \in S}{(n+2,m+n,k+m+n) \in S}$

سوال ۴

تعریف زیر را به عنوان درخت دودویی در نظر بگیرید.

$$BinaryTree ::= () \mid (Int \ BinaryTree \ BinaryTree)$$

زیردرخت هر نود را برابر درختی در نظر می‌گیریم که شامل خود آن نود به همراه نودهای زیر آن (بچه‌های آن) باشد. وزن یک نود را برابر تعداد نودهای زیردرخت آن نود می‌گذاریم. (به



مثال‌ها توجه کنید.) برنامه‌ای بنویسید که یک درخت دودویی بگیرد و عدد موجود در هر نود آن را با وزن آن نود جایگزین کند.

ورودی نمونه

```
(main '(100 () ()))
```

خروجی نمونه

```
'(1 () ())
```

ورودی نمونه

```
(main '(100 (100 () ()) (100 () ())))
```

خروجی نمونه

```
'(3 (1 () ()) (1 () ()))
```

ورودی نمونه

```
(main '(100 (100 (100 () ()) (100 () (100 () ()))) (100 () ())))
```

خروجی نمونه

```
'(6 (4 (1 () ()) (2 () (1 () ()))) (1 () ()))
```



سوال ۵

لیستی که شامل صفر یا تعدادی رشته باشد را لیست رشته‌ای می‌نامیم. رشته‌های درون یک لیست رشته‌ای را با هم همسایه می‌نامیم. برنامه‌ای بنویسید که به عنوان ورودی لیستی از لیست‌های رشته‌ای را به همراه یک رشته s دریافت کند. در نهایت لیست همسایه‌های s را خروجی دهد. لیست خروجی باید شامل تمام رشته‌هایی باشد که حداقل در یکی از لیست‌های رشته‌ای ورودی، با s همسایه بوده‌اند. ترتیب رشته‌ها در لیست خروجی مهم نیست.

ورودی نمونه

```
(main '("Fred" "A") ("Elizabeth" "B")) "Fred")
```

خروجی نمونه

```
'("A")
```

توضیح مثال: در این مثال Fred و A همسایه‌اند. Elizabeth و B همسایه‌اند. در لیست خروجی، همسایه‌های Fred آورده شده‌اند.

ورودی نمونه

```
(main '("Fred" "A") ("Elizabeth" "Betty") ("Freddie" "Fred" "F")) "Fred")
```

خروجی نمونه

```
'("A" "Freddie" "F")
```

توضیح مثال: در این مثال Fred و A همسایه‌اند. Elizabeth و Betty همسایه‌اند. Freddie و Fred و F همسایه‌اند. در لیست خروجی، همسایه‌های Fred آورده شده‌اند.

موفق باشید